Analysis of Autopsy Mobile Forensic Tools against Unsent Messages on WhatsApp Messaging Application

Fahdiaz Alief
Electrical Engineering Department
Universitas Indonesia (UI)
Depok, Indonesia
freddiziel@gmail.com

Yohan Suryanto
Electrical Engineering Department
Universitas Indonesia (UI)
Depok, Indonesia
yohan.suryanto@ui.ac.id

Linda Rosselina
Electrical Engineering Department
Universitas Indonesia (UI)
Depok, Indonesia
linda.rosselina@depok.go.id

Tofan Hermawan
Electrical Engineering Department
Universitas Indonesia (UI)
Depok, Indonesia
h3rmawantofan@gmail.com

Abstract – This paper discusses the new feature implemented in most social media messaging applications: the unsent feature, where the sender can delete the message he sent both in the sender and the recipient devices. This new feature poses a new challenge in mobile forensic, as it could potentially delete sent messages that can be used as evidence without the means to retrieve it. This paper aims to analyze how well Autopsy open-source mobile forensics tools in extracting and identifying the deleted messages, both that are sent or received. The device used in this paper is a Redmi Xiaomi Note 4, which has its userdata block extracted using linux command, and the application we're using is WhatsApp. Autopsy will analyze the extracted image and see what information can be extracted from the unsent messages. From the result of our experiment, Autopsy is capable of obtaining substantial information, but due to how each vendor and mobile OS store files and databases differently, only WhatsApp data can be extracted from the device. And based on the WhatsApp data analysis, Autopsy is not capable of retrieving the deleted messages. However it can detect the traces of deleted data that is sent from the device. And using sqlite3 database browser, the author can find remnants of received deleted messages from the extracted files by Autopsy.

Keywords—mobile forensic, social media messaging, Whatsapp, unsent feature, Autopsy, Sqlite3

I. INTRODUCTION

Nowadays, the internet can be considered a daily necessity, as we spend most of our activity online. Mobile devices, specifically smartphones, are one of the factors that enable us to do so. We could communicate with each other regardless of borders and times and shares information as soon as we heard it. We use smartphones to contact with each other, for work, getting news, getting entertainment, doing our financial transactions, and even getting rumors. All of those can be done through a smartphone, and based on Datareportal, a website that compiles data and online trends worldwide from a trusted third-party sources, the mobile user has risen 2.5% from April 2019 to April 2020, adding a total of 128 million to the total of worldwide users [1].

In Indonesia, mobile users' growth is quite significant and fast, due to more and more smartphones are affordable [2]. It also contributed to the number of people using mobile applications as a means for communication

Figure 1 Global digital growth based on Datareportal

based on Datareportal reports for Indonesia [3]. 96% of smartphone application used are chat messengers, which also represents the percentage of social media usage. Among those social media and chat messenger, WhatsApp is frequently used by Indonesian, reaching 84% usage, just below Youtube, which is the most used social media platform with 88% of usage by Indonesian.

One thing that we need to be wary of is a feature that has been garnering attention in messaging application lately, that is unsent message(s), a feature that allows the user to retract messages that are sent, whether the receiver has seen it or not.

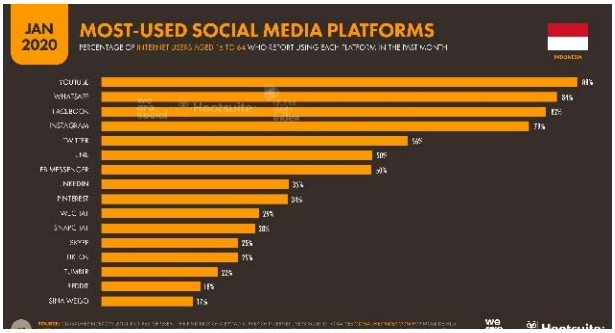

Figure 2 Indonesia most used social media

This feature has already garnered attention in Indonesia, as a case of sexual harassment between a DJ and a Youtuber. In which, when the DJ came front with the case, the Youtuber

then deleted the messages he sent from WhatsApp and other social media message with the same feature[4].

Based on the explanations above, the purpose of this paper is to study the effectiveness of open-source mobile forensic tools in extracting artifact data from the unsent feature of messaging applications. In this case, the author uses Autopsy to extract and analyze the artifacts data from WhatsApp, and analyze the effectiveness of the tools against unsent feature in WhatsApp.

II. Related Work

Mobile forensic is a field that is still in development, and due to the diverse nature of smartphone OS, extracting useful data as evidence proves to be a challenge. Over the years, several researchers have researched, both in methods and tools, to do mobile forensic.

In research by Awan [5], he showed that with only just backing up the mobile phone device, the data that is backed up can show essential information of social media accounts, which can be used to trace activities that the accounts do. Although this method is limited because it depends on how the mobile phone back up its data, from Awan's research, he cannot get any useful information from blackberry mobile phones, unlike iOS and Android mobile phones.

Another extensive research for IMO chat messenger done by Ababneh et al.[6] shown that by using autopsy and Sqlite browsers, they can find traces of messages and calls done by the application after extracting information using Autopsy and viewing the databases with Sqlite.

A forensic analysis of Line, Kakao, and Telegram was conducted by Satrya et al. [7]. Their research is capable to find several artifacts from those three application by manually analyzing the artifacts using Sqlite Browser and several other tools. The result of their research is that they're able to recover normal and private message sent, although they can't recover deleted messages.

Adwan et al. [8] also researched WhatsApp unsent feature. In their research, using manual digital forensic and analyzing the artifacts left by WhatsApp manually, they found out that WhatsApp didn't log the change that occured because of the unsent feature. Because of that, Adwan and their team concluded that the unsent feature isn't in favor of preserving evidence against cybercrime using unsent feature.

Research by Shortall et al. [9] that also research WhatsApp artifact messages using UFED and Oxygen forensic tools showed that they could recover messages, pictures, and contact info on android and iOS devices. But they cannot do the same to windows phone due to inaccessible encryption key.

A research by Mirza et al. [10] propose a solution for the unsent feature, which is creating an application that reads and copies the WhatsApp messages received through its notification. Mirza and their team acknowledge that this solution is more of a brute force solution, which could potentially create another security risks, as they also mentioned that there're several applications that uses the same method they used for their application. So instead, they proposed several changes to WhatsApp unsend feature.

Comparative research of both open source and paid forensic tools is done by Padmanabhan et al. [11]. In their study, the tools they compared gave varying result, and each open source and paid tools have their strength and weakness. So they concluded that a "one-size-fits-all" is not the correct approach in choosing a forensic tool, but instead using any tools that fit what we need. Another comparative research of forensic tools is done by Sathe et al. [12], but they compared the tools based on the acquisition method, which is physical and logical acquisition method. From their research, they reached the same conclusion with [11], which is no single tool can provide complete insight of the device analyzed, so it's better to use several tools.

III. Forensic Tools and Methodology

In this experiment, the author uses a Xiaomi Redmi Note 4 phone that is used to extract and analyze the image. The phone has 64 GB of internal storage space without an SD card. The image extracted would be the decrypted userdata block that holds information on all applications installed on the phone. To make sure that the image and data extracted from the image can be used in a court of law, the author follows SWGDE Best Practice for Mobile Device Evidence Collection & Preservation, Handling, and Acquisition [13].

- **Preserving the mobile device**, SWGDE mentioned that we would need to block any incoming transmission that can alter the current data condition of the device by utilizing airplane mode. If we can't use airplane mode or it's not enough, we can keep the mobile device in a faraday bag. In this paper, the device used is set to airplane mode when extracting the image, so no transmission can be received or sent by the device.
- **Creating the image of the android for analysis**, the extraction method that the author uses can be considered an extraction of file systems by SWGDE, as the image created is a block of the file system that contains the data that needs to be analyzed. SWGDE also mentions the necessity of using a bootloader or gaining root access to enter the file system needed. Because the methods of rooting a device can leave behind a digital artifact, it should be documented if one of those methods is used.

The experiment scenario used is similar to the case mentioned above, in that the author will send a message and then unsent it. But the image extraction is done after several days, in this case, a week, to see how much data we can get from the extracted image. As in real life, we would need to wait for clearance to do so.

For the forensic tools, Autopsy is chosen as the tool for analyzing the extracted image, as Autopsy is capable to both analyze and export the data inside the extracted image. So if further investigation is needed, we can extract the data and use more suitable tools or methods for investigating furthermore.

Below explains step by step what the author did from extracting the image to analyzing the extracted image.

A. *Rooting the phone*

To extract the image needed for analysis, in general, we need to root the phone to be able to access the */data* folder and create an image of it. The author will not explain in detail on how to root the phone as it depends on the brands and vendors of the phone, but there are two ways that a phone can be rooted. The first one is using an application called KingoRoot that is an app that can give root access without the need to modify your phone ROM, so it's the safest way, and you can easily remove the root access after finished. Kingoroot is using a certain way to grant root access to a smartphone, but, this method is not guaranteed to succeed because of how each vendor has different protection for application from gaining root access, there's a chance that your phone cannot be rooted this way, so we use the second way, which is flashing your phone and installing the root access. This method has a higher chance of succeeding and a higher risk of bricking your phone, so please find info about rooting your phone and also how to unbrick your phone if something goes wrong when using the flashing method.

B. *Extracting the image*

After rooting your phone, and confirming that you have root access, we can move to the next step, which is extracting the image. Before we start extracting the images, we need to install busybox, which will install a set of linux commands to our phone, because the linux command we need in android is limited or missing.

After busybox is installed, we will need to enable development features, USB debugging and then use ADB either from the Android SDK or Android Studio in the investigator pc. Connect our phone to the investigator pc, open up a command shell in the ADB folder and check for device connection with this command

adb devices

It should run the ADB daemon and detects your connected phone. After that, run the command below

adb shell

The command will enter our phone via command shell. Next, we need to change into the root user and see the partition we need. Generally, what we need is inside the *userdata* block, so we have two options, we can image the whole disk block, or we can just image the *userdata* block. For minimizing storage usage in investigator pc or the mobile phone extra storage like an SD card, we will only image the *userdata* block. First, we need to know what block *userdata* partition is linked with, so we use this command as root:

Ls –al /dev/block/by-name/

The command above will show you a lists of filesystem block with its linked partition, these lists will tell you which filesystem block is used as *userdata*. Since this device doesn't have an SD card, we can't save the image on the phone itself as it will exceed the storage space, so we will send it to the investigator pc instead. To do that, we need to open another terminal in the ADB folder. We need to open the port for transfer first, using this command:

adb forward tcp:8888 tcp:8888

Now we can send the image through that port to the investigator pc directly. To do that, we will use this command:

Dd if=/dev/block/[the userdata block] | busybox nc –l –p 8888

The dd command will start to image that block, and the pipe symbol (|) will transfer the image to the nc command. Nc command is not available in android so that is why we install and use busybox, nc will then send the image through port 8888 to our investigator pc. After that, we need to receive the image file in the investigator pc. To do that, use this command below:

nc 127.0.0.1 8888 > [imagefile_name.image_extension]

Nc command is not available in Windows, but you can easily find the .exe files needed to run the command in Windows. The image extension type that can be used and known for Autopsy is .dd and .img, so you can change the [] part with something like *userdata.dd* or *userdata.img*. After that, wait until the image finish transferring.

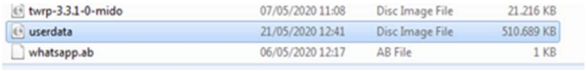

Figure 3 Extracted userdata image

After the image is extracted, we can insert it into Autopsy to be analyzed. The GUI in Autopsy is easy to use. Hence the process is very straight forward, we just need to create a new case and insert the extracted image as the data source. There're a lot of modules that can be used not just for Android image analysis, so only enable Android Analyzer ingest module to shorten analysis time.

IV. RESULT ANALYSIS

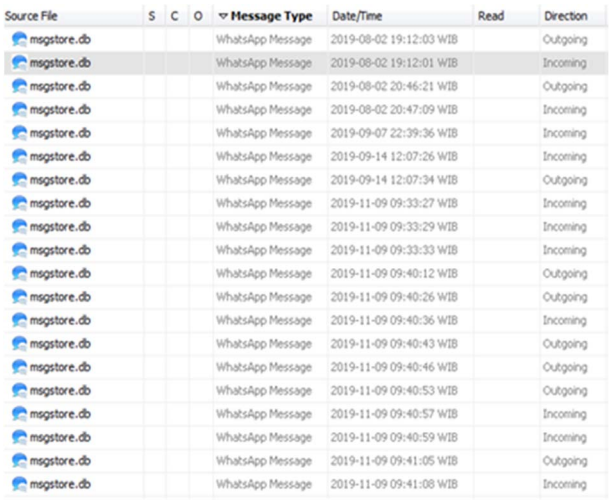

Figure 4 Extracted and identified messages by Autopsy

After the ingest module finished analyzing the image, Autopsy will automatically extract the data to be analyzed by itself. There are several pieces of information that can be read from the data, as mentioned in the user documentation, Autopsy can extract data from several messaging

applications, including WhatsApp [14]. But because each vendor has a different OS, the way they keep the application databases varies for each application, so Autopsy cannot cover all OS versions and vendors [15]. In this paper's case, the only extracted application data is only from WhatsApp, so Autopsy fits the author's need.

There're in a total of 100376 extracted messages, and that is mostly WhatsApp messages, as shown in figure 4. To simplify the need to find the correct message, the author creates a test communication to unsent a message from one contact and then finds it in Autopsy.

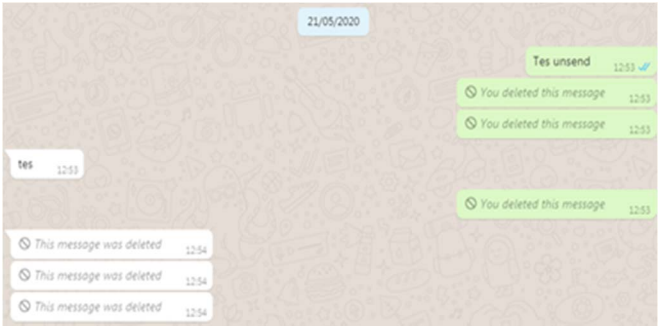

Figure 5 Test communication messages

The messages in Fig. 5 shows a test communication, each unsent three messages. In Autopsy, we can easily navigate communications that happened with the communication windows. Below are the same messages that are found in Autopsy using the communication windows:

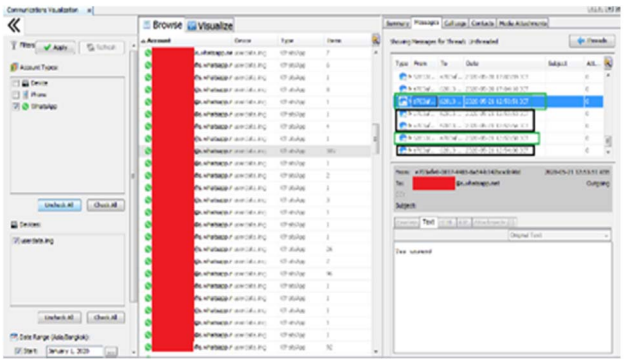

Figure 6 Autopsy communication window

The messages are filtered to only show WhatsApp Messages and from the first January of 2020. The grey highlighted account is the one that the author uses to send and receive the messages then unsent it. As you can see on the right side of the window, the messages that the author sent showed up, and the item that contains it is highlighted with the green box. The black box highlighting three items is the message that the author unsent. Unlike the rest of the items, which contains a text and/or attachment info, the unsent message has no content. As seen in Fig. 6, the content inside the items is empty, as highlighted by the red box. The interesting thing is that Autopsy keeps the last text shown by an item when selecting the unsent message, which is shown by the green box. Another point of interest is that only Outgoing messages the author sent then unsent leaves an artifact while the messages that the author received then unsent are not showing up or not detectable by Autopsy.

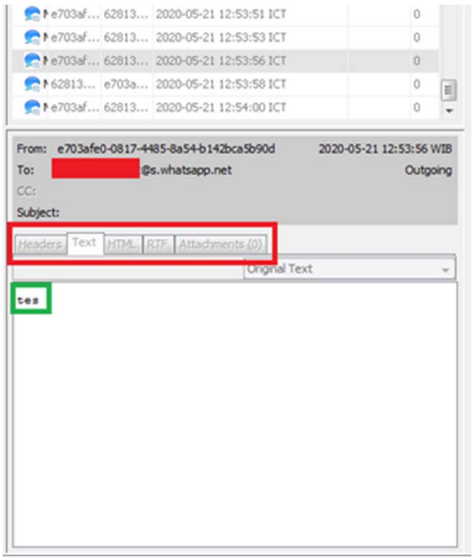

Figure 7 Unsent message content

So with that in mind, since Autopsy is capable of extracting files from the images that it uses for analyzing, the author extracts the WhatsApp database and tries to find the unsent messages or artifacts from inside the database. To extract files, we can easily navigate to the location of the files in the images using the context menu on the message source file location column, as shown in Fig. 7. And then, we can easily extract the files using the context menu again.

From there, the author is using sqlite3 to browse the extracted database. From the database itself, several interesting tables is worth looking into, and they are *deleted_chat_job, deleted_messages_ids_view, deleted_messages_view, message, message_revoked*, and *messages*. Aside from *messages*, most of the tables mentioned are empty, so we will take a look inside the messages table. To be able to view and modify the table so it can be easily viewed, the author created a .CSV file output of the table using the command below:

```
sqlite3.exe "[database location]"
.mode list
.separator ,
.output messages.csv
select * from messages;
.exit
```

Fig. 8 below shows the important items inside the message table. Some headers and their items are omitted to make it brief and relevant to this paper.

id	key_remote_jid	key_from me	data
129554	62813xxxx@s.whatsapp.net	1	Tes unsend
129555	62813xxxx@s.whatsapp.net	1	
129556	62813xxxx@s.whatsapp.net	1	
129557	62813xxxx@s.whatsapp.net	0	tes
129558	62813xxxx@s.whatsapp.net	1	
129559	62813xxxx@s.whatsapp.net	0	
129560	62813xxxx@s.whatsapp.net	0	
129561	62813xxxx@s.whatsapp.net	0	

Figure 8 messages table content

As seen above, the *message* data is deleted from the database itself, but it still contains a record of the message being sent and received. The message is the same as shown in Fig. 7, with three received and sent messages being unsent. Unfortunately, the author is unable to recover the contents of the unsent message.

V. Conclusion

Autopsy is a powerful open-source tool for digital forensic; not only it's able to analyze the data from the image given but also capable of extracting the files from the device image to be used with other forensic tools for in-depth analysis. Unfortunately, Autopsy belongs to logical system acquisition. Therefore, if any file is deleted and/or hidden to the filesystem, Autopsy may not be able to recover it, unlike other digital forensic tools that are categorized as physical acquisition, which ignores what the filesystem can find and just find out themselves. The reasoning above is what the author think why Autopsy cannot recover the unsent messages because it has been deleted in WhatsApp database itself. However, the artifacts that is found by Autopsy proves that it can be used as an evidence-gathering tools, as long as the image that is extracted by the investigator is preserved according to SWGDE standards, or any data extraction models/frameworks standards.

As it also has been mentioned before, due to how different mobile phone OS and vendors are, getting artifacts from some mobile applications that is supported and can be identified by Autopsy are not reliable due to different database location, because of varying mobile phone OS and vendors. So analysis using Autopsy for mobile forensics might be a bit finicky. However, it is an excellent tool to extract and browse data files in mobile data images for preliminary investigations and analysis. As [9] has mentioned in their paper, it's better to use one or several forensic tools that fit your needs, or as [5] has done too, we might not need to use other forensic tools and use a simple database or file browser to search from backed file of our phone or the extracted files from Autopsy, like what the author of this paper did.

For future works, the author strongly recommends for developing a way to retrieve these unsent messages, whether with a new or improving current Autopsy modules, or simply using other tools that are also open-source which uses physical instead of logical acquisition. And as mentioned above, researching for a way to make Autopsy able to find the artifacts from the application that it supports regardless of smartphone OS and vendors is also another good idea to research. Simpler research that could be done is to use a different scenario for extracting and analyzing the image data, as the author only extracts the mobile phone image and analyzes the data after several days has passed. Another thing worth researching is defined protocols and frameworks to analyze and extract the data we needed without the use of forensic tools and ways to preserve this extracted data so it can be used as evidences.

References

[1] https://datareportal.com/global-digital-overview
[2] https://www.emarketer.com/Article/Smartphones-Move-Upmarket-Indonesia/1016459T
[3] https://datareportal.com/reports/digital-2020-indonesia
[4] https://wow.tribunnews.com/2019/09/30/atta-halilintar-disebut-lupa-hapus-1-pesan-dm-ke-bebby-fey-begini-isinya
[5] Awan, F. A., "Forensic Examination of Social Networking Application on Smartphones", in 2015 Conference on Information Assurance and Cyber Security (CIACS)
[6] Ababneh, A., Awwad, M. A., Al-Saleh, M. I., "IMO Forensics in Android and Windows Systems"
[7] G. B. Satrya, P. T. Daely, and S.Y. Shin, "Android Forensic Analysis: Private Chat on Social Messenger", in *Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, Vienna, Austria, 2016
[8] S. Adwan and F. Salamah, "A Manual Mobile Phone forensic approach towards the analysis of WhatsApp Seven-Minute Delete Feature", in *21st Saudi Computer Society National Computer Conference (NCC)*, Riyadh, Saudi Arabia, 2018
[9] Shortall, A., Bin Azhar, M. A. H., "Forensic acquisitions of WhatsApp data on popular mobile platforms", in *2015 Sixth International Conference on Emerging Security Technologies*
[10] M. M. Mirza, F. E. Salamh, and U. Karabiyik, "An Android Case Study on Technical Anti-Forensic Challenges of WhatsApp Application", in *21st Saudi Computer Society National Computer Conference (NCC)*, Riyadh, Saudi Arabia, 2018
[11] R. Padmanabhan et al,"Comparative Analysis of Commercial and Open Source Mobile Device Forensic Tools", in *Ninth International Conference on Contemporary Computing (IC3)*, Noida, India, 2016
[12] S. C. Sathe and N. M. Dongre, "Data Acquisition Techniques in Mobile Forensics", in *2nd International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, India, 2018
[13] Best Practices for Mobile Device Evidence Collection & Preservation, Handling, and Acquisition, SWGDE, 2019
[14] https://github.com/sleuthkit/autopsy/releases/
[15] https://sleuthkit.org/autopsy/docs/user-docs/4.8.0/android_analyzer_page.html